INSTITUT FÜR INFORMATIK

A Kleene Theorem for Regular Picture Languages

Oliver Matz

Bericht Nr. 0703
February 16, 2007

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

KIEL

Institut für Informatik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

A Kleene Theorem for Regular Picture Languages

Oliver Matz

Bericht Nr. 0703
February 16, 2007

e-mail: matz@ti.informatik.uni-kiel.de

A Kleene Theorem for Regular Picture Languages

Oliver Matz

Institut für Informatik, Universität Kiel, Germany

matz@ti.informatik.uni-kiel.de

February 16, 2007

Abstract

We consider the class of regular picture languages, which is the two-dimensional analog to the well-known class of regular (word) languages as defined by regular expressions. We present an automaton model that characterizes the class of regular picture languages, similar to the famous Kleene theorem that characterizes the class of regular picture languages by non-deterministic finite state automata.

The two concatenations (called "row"- and "column"-concatenation) for picture languages are partial because they require matching height or width, respectively. However, we show that for the definition of regular picture languages, the partialness of the concatenations is irrelevant in the sense that every regular expression can be converted into a regular expression during whose evaluation no picture gets discarded because of that partialness.

We conclude that for every regular picture languages, its "front" (i.e., the set of top rows of its pictures) is a regular word language.

1 Introduction

Considerable effort has been made in the recent years to investigate which results from well-known formal language theory can be preserved when considering pictures (i.e., arrays) rather than words. See for example [GR96, GRST96, Wil97, LS97, AGM04, AGM05, AM05, Mat97, Mat98]. Most authors concentrate an the transfer of the results about the class of regular word languages. The notion of *tiling-systems* has been introduced and is usually considered the natural adaption of the notion of non-deterministic finite state automata to pictures. This is because there are several equivalent variants of this notion, see [IN77, LS94, GR96] and the class has relatively strong closure properties. The class of picture languages recognized by tiling-systems is called *tiling-recognizable*.
However, there is a number of reasons why I think that this class is "too large". Firstly, it contains picture languages that should not be called "recognizable". For example, for every linear bounded automaton $\mathfrak{A}$ (a Turing machine that never exceeds its input area), the set of pictures that (in the straightforward way) encode a run of $\mathfrak{A}$ is tiling-recognizable.

This implies in particular that the emptiness problem is undecidable for tiling-systems. Secondly, the membership problem is NP-complete. Thirdly, it contains the language of all squares over $\{a, b\}$ with equally many a's and b's as well as some other languages that do not "feel finite-state", see [Rei98]. Fourthly, it is far larger than the class of *regular* picture languages as defined by regular expressions (or, equivalently, by closure properties, see below).
There is a natural interest in a Kleene-like theorem for picture languages. This task is usually understood as: strengthen the power of regular expressions so that they capture tiling-systems in expressive power. One easy way to achieve this is to add intersection as well as alphabet projection to the regular expression side, i.e., one considers the class $\mathrm{REG}_\cap(\Sigma)$ of picture language over Σ that contains the singleton languages and is closed under row- and colum concatenation and -closure as well as under union and intersection. Then an easy (and not Kleene-like) proof shows that for every tiling-system T there exists a (potentially very large) alphabet Γ and a language $L \in \mathrm{REG}_\cap(\Gamma)$ such that the picture language recognized by T is the image of L under some alphabet projection defined on Γ. However, having both intersection and projection destroys the assembling character of regular expressions, so the above is not really a Kleene-like theorem.

Position Pushdown Automata

We consider the class REG of picture languages over Σ that contains the singleton languages and is closed under row- and colum concatenation and -closure as well as under union.
In my opinion, this class REG of regular picture languages is very promising for practical applications because its emptiness- and membership problem are decidable in polynomial time and it is simple enough to be understood by humans.
As far as I know, there is no Kleene-like theorem[1] for the class of regular picture languages, i.e., no automaton model that captures this class. In this paper, we investigate such an automation model, the so-called *position pushdown picture automata* (PPPA). We start with an informal description of PPPA and their operation. Formal definitions can be found in the next section.

How a PPPA works When a PPPA scans an input picture, it starts with its head placed on the input symbol in the top left corner. The PPPA proceeds stepwise, scanning each cell exactly once.
After each step, the "input-front" between those cells that *have been* scanned and those that *will be* scanned is given by a zig-zag line from the left or bottom edge right-upwards to the right or top edge of the input picture (see Figure 1).
This front is established by one or more cells that are the "inside-corners" right from and below that line, one of which is the head position. In Figure 1, there are four such inside-corners: the head position is marked by a "$\times$" and the other ones by a "$\bullet$".

[1][Mat95] suggests an adaption of right-linear grammars from word languages to picture languages. These grammars capture regular picture languages in expressive power, which might be interpreted as a Kleene-like theorem. However, these grammars do not translate to an automaton model as easily as the right-linear grammars do.

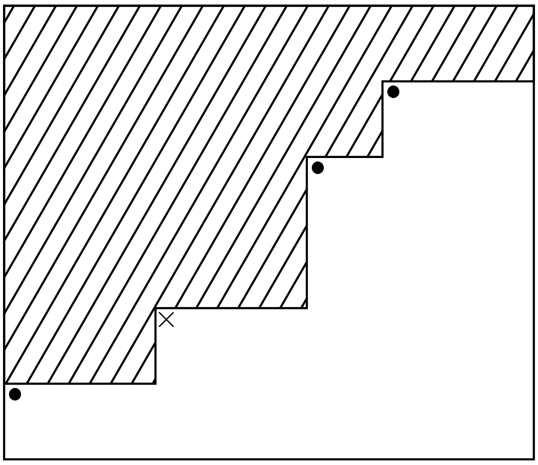

Figure 1: The "input-front" of a picture while being scanned by a PPPA. The hatched area has already been scanned.

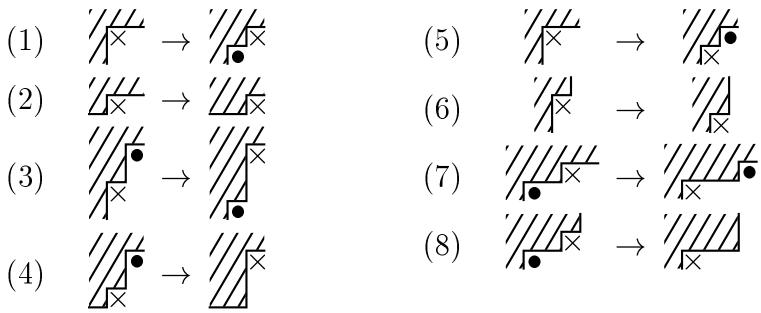

Figure 2: The possible local modifications of the input-front.

In each step, the PPPA scans the symbol of the cell that its head is placed on, thereby adding that cell to the scanned part right/above the input-front. Then it places its head on a new inside-corner close to the former head position.
By this modification, the number of inside-corners may be either incremented, decremented, or left unchanged. Figure 2 sketches the possible input-front modifications the PPPA performs during a single step. The zig-zag line indicates a fragment of the input-front, the cross at the dots indicate the head position and neighbored inside-corners.
Since these local modifications of the input-front depend only on the head position and the two neighbored inside-corners of the input-front, we may think of these inside-corners as being stored on two stacks: the left stack stores the inside-corners left and below the head, whereas the right stack stores the inside-corners right and above the head. The stack top of each stack is its inside-corner closest to the head.
Just like ordinary NFA, a PPPA has a finite set of states to control its operation. A configuration of a PPPA is given by the input front together with an assignment of inside-corners to states, i.e., the mentioned two stacks store one state together with each inside-corner.
In order to ensure that the PPPA do not exceed the class of regular picture languages in

expressive power, we have to add the following constraint to the definition of PPPA. There must be a quasi-ordering on the set of states, and the states on the two stacks at a given time are also strictly ordered by this quasi-ordering.
Among the symbols on the two stacks, the one with the smallest state is always one of the two stack tops and it is the only one that may be popped. This way, these two stacks behave very much like a single one.
See Definition 2.3 below for a formal definition of the step relation.

Pictures, Periodicity, and Alternation

Apart from the automata, this paper contributes also a few results about the structure of the class of regular picture languages. It has been shown in [Mat97] that for every regular picture language, the set of sizes of its pictures (as a set of pairs of natural numbers) is a finite union of Cartesian products of ultimately periodic subsets of $\mathbb{N}$.
It is well-known that the class of regular picture languages is not closed under intersection. However, in this paper we show that for every regular picture language L and every Cartesian product X of ultimately periodic subsets of $\mathbb{N}$, the set of all pictures of L whose size is in X is regular. We conclude that the partialness of the concatenations is not essential for the definition of regular picture languages.

2 Definitions

2.1 Pictures and Picture Languages

Throughout the paper, we consider a fixed finite alphabet Σ. A *picture P over Σ* is an array over Σ, i.e., a mapping of the form $\{0,\ldots,m-1\}\times\{0,\ldots,n-1\}\rightarrow\Sigma$ with $m,n\geq 1$. For such a picture, m (or n) are called the *height* (or *width*, respectively) and are denoted by $\overline{\underline{P}}$ (or $|P|$, respectively). We identify a word over Σ with a picture of height 1 in the straightforward way. A *picture language (over Σ)* is a set of pictures over Σ.
Although it is common practice and sometimes convenient, we do not consider an "empty picture" that serves as a neutral element in both concatenations.
We define two partial concatenations on the set of pictures. For two pictures P, R over Σ, the *column concatenation* $P \varobar R$ (or *row concatenation* $P \ominus R$, respectively) is defined iff $\overline{\underline{P}} = \overline{\underline{R}}$ (or $|P| = |R|$, respectively) by the result of juxtaposing R to the right (or to the bottom, respectively) of P. The result is a picture of length $|P| + |R|$ (or of height $\overline{\underline{P}} + \overline{\underline{R}}$, respectively).
As usual, these partial concatenations may be lifted to total operations on picture languages. These operations may be iterated as follows: For a picture language L, the *column closure* $L^{\varobar+}$ (or the *row closure* $L^{\ominus+}$, respectively) is defined as the smallest picture language that is a superset of L and is closed under column concatenation (or row concatenation, respectively).
The class of *regular picture languages over Σ* (denoted REG(Σ)) is the smallest class of picture languages over Σ that contains all singleton languages whose element is a 1×1-picture and that is closed under row concatenation, column concatenation, row closure, column closure, and union.

This way, the class of regular picture languages is the natural two-dimensional extension of the class of regular picture languages. In this paper, we do not consider syntactic notions such as regular picture expressions, so there is no need to introduce them here, although this is straightforward.

2.2 Automata on Pictures

For a given quasi-ordering $\leq$ we will always write $\equiv$ for the equivalence relation $\leq \cap \geq$, and set $< := \leq \setminus \equiv$. If $\leq$ is a quasi-ordering on the set Q and $q \in Q$, then $[q]_{\equiv}$ (or $[q]$ for short) denotes the equivalence class of q wrt. $\equiv$.
The two distinct symbols $\mathbf{d}, \mathbf{r}$ represent the two directions down and right. A *position pushdown picture automaton* (PPPA for short) is a tuple $(Q, \Sigma, q_0, \Delta, F, \sigma, \leq)$, where

- Q is a finite set of *states* not containing the special symbol $\top$. We write $\widehat{Q} = Q \cup \{\top\}$.
- $\leq$ is a reflexive, transitive relation on Q. It is extended to $\widehat{Q}$ by defining $q \leq \top$ for all $q \in \widehat{Q}$.
- $q_0 \in Q$ is the *initial state*,
- $F \subseteq [q_0]$ is the set of *final states*,
- $\sigma : Q \rightarrow \{\mathbf{d}, \mathbf{r}\}$
- $\Delta \subseteq Q \times \Sigma \times \widehat{Q} \times \widehat{Q}$,

with the following conditions:

- For all $(q, a, d, r) \in \Delta$ we have $d < r \vee r < d$.
- $\sigma(q) = \sigma(q')$ for all $q, q' \in Q$ with $q \equiv q'$.
- For all $q, q' \in Q$ there is a $q'' \in Q$ with $q, q' \leq q''$.
- q_0 is maximal in Q wrt. $\leq$.

For technical convenience, we need to extend the domain of a picture. The *extended domain of* P is $\mathrm{dom}'P = \mathrm{dom}P \cup \{(\underline{\overline{P}}, -1), (-1, |P|)\}$.
Let P be a picture. Consider the alphabet $\Gamma = \{(q, i, j) \mid q \in Q, (i, j) \in \mathrm{dom}P\} \cup \{(\top, -1, |P|), (\top, \underline{\overline{P}}, -1)\}$. For a symbol $(q, i, j) \in \Gamma$ we define $\mathrm{state}(q, i, j) = q$, and $\mathrm{col}(q, i, j) = i$, and $\mathrm{row}(q, i, j) = j$. In the context of stack content, the first symbol of a non-empty word α is denoted $\mathrm{top}(\alpha)$ and referred to as *stack top*. The last symbol is referred to as *stack bottom*.
For a word α over Γ, we define

$$\mathrm{state}(\alpha) = \begin{cases} \mathrm{state}(\mathrm{top}(\alpha)) & \text{if } \alpha \neq \varepsilon \\ \top & \text{else} \end{cases}$$

$$\mathrm{row}(\alpha) = \begin{cases} \mathrm{row}(\mathrm{top}(\alpha)) & \text{if } \alpha \neq \varepsilon \\ \infty & \text{else} \end{cases}$$

$$\mathrm{col}(\alpha) = \begin{cases} \mathrm{col}(\mathrm{top}(\alpha)) & \text{if } \alpha \neq \varepsilon \\ \infty & \text{else} \end{cases}$$

Definition 2.1 A *P-configuration of* $\mathfrak{A}$ is a tuple $(P, h, \alpha_m \dots \alpha_1, \beta_n \dots \beta_1)$, where $m, n \geq 0$ and $h, \alpha_m, \dots, \alpha_1, \beta_n, \dots, \beta_1 \in \Gamma$ with the following properties

$$\begin{array}{ccccccccc}
\text{state}(h) & < & \text{state}(\alpha_m) & < & \dots & < & \text{state}(\alpha_1) & \leq & \top \\
\text{row}(h) & < & \text{row}(\alpha_m) & < & \dots & < & \text{row}(\alpha_1) & \leq & \overline{\underline{P}} \\
\text{col}(h) & > & \text{col}(\alpha_m) & > & \dots & > & \text{col}(\alpha_1) & \geq & -1 \\
\text{state}(h) & < & \text{state}(\beta_n) & < & \dots & < & \text{state}(\beta_1) & \leq & \top \\
\text{row}(h) & > & \text{row}(\beta_n) & > & \dots & > & \text{row}(\beta_1) & \geq & -1 \\
\text{col}(h) & < & \text{col}(\beta_n) & < & \dots & < & \text{col}(\beta_1) & \leq & |P|
\end{array}$$

$$\begin{array}{rl}
\text{If } m \geq 1, \text{ then:} & \text{state}(\alpha_1) = \top \leftrightarrow \text{row}(\alpha_1) = \overline{\underline{P}} \leftrightarrow \text{col}(\alpha_1) = -1 \leftrightarrow \sigma(q_0) = \mathbf{d} \\
\text{If } n \geq 1, \text{ then:} & \text{state}(\beta_1) = \top \leftrightarrow \text{row}(\beta_1) = -1 \leftrightarrow \text{col}(\beta_1) = |P| \leftrightarrow \sigma(q_0) = \mathbf{r}
\end{array}$$

□

Figure 3 sketches a few configurations that an automaton might reach during a run, in particular the ordering of the stack symbols in the above definition.
Intuitively, q is the current state and (i, j) is the current head position of the PPPA. The two stacks control a search of the PPPA. The meaning is the following: Any cell (i, j) has already been visited iff there is a symbol $\gamma \in \{h, \alpha_m, \dots, \alpha_1, \beta_n, \dots, \beta_1\}$ such that $(i \leq \text{col}(\gamma) \wedge j < \text{row}(\gamma)) \vee (i < \text{col}(\gamma) \wedge j \leq \text{row}(\gamma))$.

Remark 2.2 Let $\mathfrak{A}$ be a PPPA. Let $\kappa = (P, (q, i, j), \alpha, \beta)$ be a configuration with $q \in [q_0]$ and $\sigma(q_0) = \mathbf{r}$. Since q is maximal in Q wrt. $\leq$, Definition 2.1 implies $i = 0$ and $\beta = \varepsilon$. In other words, the states in $[q_0]$ can only be used in the top row. □

The *initial P-configuration* of $\mathfrak{A}$ is

- $(P, (q_0, 0, 0), (\top, \overline{\underline{P}}, -1), \varepsilon)$, if $\sigma(q_0) = \mathbf{r}$, or
- $(P, (q_0, 0, 0), \varepsilon, (\top, -1, |P|))$, if $\sigma(q_0) = \mathbf{d}$.

A *final P-configuration* of $\mathfrak{A}$ is of the form

- $(P, (f, 0, |P|), (\top, \overline{\underline{P}}, -1), \varepsilon)$, if $\sigma(q_0) = \mathbf{r}$, or
- $(P, (f, \overline{\underline{P}}, 0), \varepsilon, (\top, -1, |P|))$, if $\sigma(q_0) = \mathbf{d}$,

for some final state $f \in F$. In other words, a PPPA has reached a final configuration if it has scanned the whole input, reestablished the initial stack contents, and placed the head just outside the picture in the top row (or left column, respectively).
Me may now proceed with the formal definition of how a PPPA modifies its configuration in one of the ways sketched in Figure 2.

Definition 2.3 Let $\mathfrak{A}$ be a PPPA and P be a picture over Σ. The *step relation* $\vdash$ *of* $\mathfrak{A}$ on the set of P-configurations is defined as follows. Let $\kappa = (P, (q, i, j), \alpha, \beta)$ be a

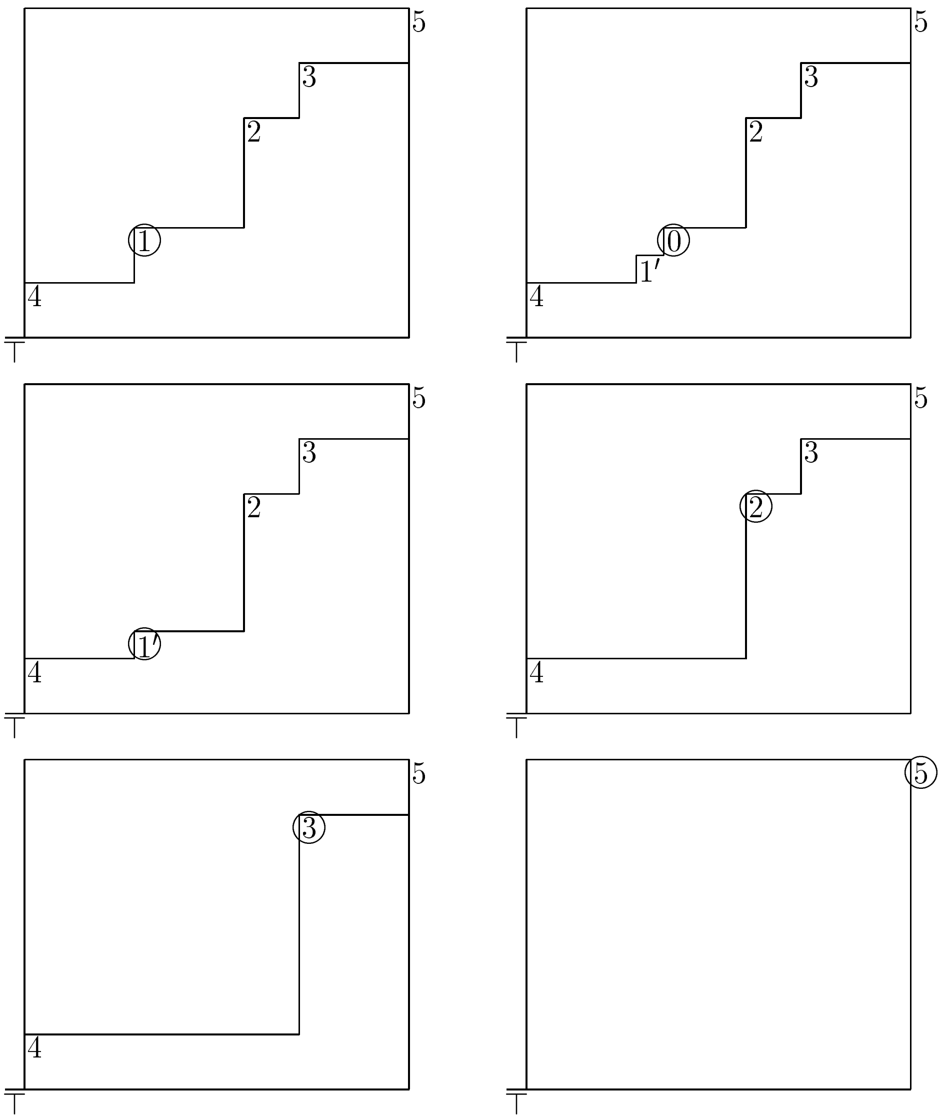

Figure 3: Some intermediate configurations of a PPPA with $0 < 1 \equiv 1' < 2 < 3 < 4 < 5 \equiv q_0$ and $\sigma(0) = \sigma(2) = \sigma(3) = \sigma(5) = \sigma(q_0) = \mathbf{r}$ and $\sigma(1) = \sigma(4) = \mathbf{d}$. The current state is marked by a circle. The states are popped from the two stacks in ascending order. The last configuration is a final configuration in case 5 is a final state.

configuration and $(q, P(i,j), d, r) \in \Delta$ be a transition. Set

$$\alpha' := \begin{cases} (d, i+1, j)\alpha & \text{if } d < \text{state}(\alpha) \wedge i+1 < \text{row}(\alpha), \\ \alpha & \text{if } d = \text{state}(\alpha) \wedge i+1 = \text{row}(\alpha), \\ \bot & \text{else;} \end{cases}$$

$$\beta' := \begin{cases} (r, i, j+1)\beta & \text{if } r < \text{state}(\beta) \wedge j+1 < \text{col}(\beta), \\ \beta & \text{if } r = \text{state}(\beta) \wedge j+1 = \text{col}(\beta), \\ \bot & \text{else.} \end{cases}$$

The successor configuration κ' of κ under transition $(q, P(i,j), d, r)$ is defined iff both α' and β' are defined, namely as follows:
If $\text{state}(\alpha') < \text{state}(\beta')$, then decompose α' as $\alpha' = h'\alpha''$ and set $\kappa' := (P, h', \alpha'', \beta')$.
If $\text{state}(\alpha') > \text{state}(\beta')$, then decompose β' as $\beta' = h'\beta''$ and set $\kappa' := (P, h', \alpha', \beta'')$.
We write $\kappa \vdash \kappa'$ iff κ' is the successor configuration of κ under some transition. As usual, $\vdash^*$ denotes the reflexive and transitive closure of $\vdash$. □

The step relation indeed preserves the invariant for configurations stated in Definition 2.1. Intuitively, the meaning of a transition (q, a, d, r) is the following: When in state q and the current cell contains letter a, then two stack symbols are created: Firstly, the bottom neighbor cell is combined with d, and secondly, the right neighbor cell is combined with r. Then a specific push operation is performed that may temporarily push the first symbol on the left stack and the second on the right stack. Such a push-operation fails if the fresh symbol "overtakes" the former stack top of its stack. However, such a push-operation causes the fresh symbol to silently vanish if it matches the former stack top, and it is an ordinary push operation if is compatible with the former stack content.
To illustrate this, please refer to Figure 2: the fresh symbol for the left stack vanishes in patterns (2), (4), (7), and (8); the fresh symbol for the right stack vanishes in pattern (3), (4), (6), and (8).
After these zero, one, or two temporary stack modifications, the smaller of the two stack symbols is popped from its stack and provides for a new state and head position. Note that this head position may, but need not, be a neighbor cell of the previous head position. In Figure 2, the smaller stack symbol is provided by the right stack in patterns (1)-(4) and by the left stack in patterns (5)-(8)
This way, the automaton proceeds. In order to accept its input, it must terminate either by moving its head onto the topmost cell beyond the right edge of the picture (in case $\sigma(q_0) = \mathbf{r}$) or onto the leftmost cell beyond the bottom edge of the picture (in case $\sigma(q_0) = \mathbf{d}$), in each case reaching a final state.
A sequence of P-configurations $\kappa_0 \vdash \ldots \vdash \kappa_m$ is called a *run on P* if κ_0 is initial and κ_m is final. We say that the PPPA $\mathfrak{A}$ *accepts* a picture P if there is a run on P. The set of all pictures accepted by $\mathfrak{A}$ is called the picture language *recognized by* $\mathfrak{A}$ and is denoted $L(\mathfrak{A})$.
In the remainder, we will often have two or three PPPAs under consideration and use decorations to distinguish them. By convention, if the PPPA is denoted e.g. $\mathfrak{A}'$, then q_0' denotes its initial state. Likewise, the corresponding notions $\leq'$, $\equiv'$, $[q]'$, Δ', $\vdash'$ etc. are then implicitly defined with their obvious meanings. The analogous convention applies if the PPPA is named $\mathfrak{A}_1$ or the like.

Remark 2.4 For the operation of a PPPA, the content of formerly scanned or non-scanned input does not affect its operation: Suppose $\sigma(q_0) = \mathbf{r}$ and a picture of the form $P = P_1 \varobar P_2 \varobar P_3$ and suppose

$$(P, (q, 0, |P_1|), (\top, \underline{P}, -1), \varepsilon) \vdash^* (P, (p, 0, |P_1 \varobar P_2|), (\top, \underline{P}, -1), \varepsilon),$$

i.e., when started with initial stack contents at the top left cell of the "infix" picture P_2, the PPPA $\mathfrak{A}$ consumes that infix picture completely and thereby re-establishes the initial stack contents and reaches state p. Let P_1', P_3' be pictures of same height as P and let $P' = P_1' \varobar P_2 \varobar P_3'$. Then we have

$$(P', (q, 0, |P_1'|), (\top, \underline{P'}, -1), \varepsilon) \vdash^* (P', (p, 0, |P_1' \varobar P_2|), (\top, \underline{P'}, -1), \varepsilon).$$

This is because by definition of the step relation, the head can never visit a cell in $\mathrm{dom}(P_1)$ or outside $\mathrm{dom}(P_1 \varobar P_2)$ in the intermediate steps of the above chains of configurations. The statement remains true if one or more of $P_1, P_2, P_3, P_1', P_3'$ are "empty pictures", i.e., vanish. (Here is a point in this paper where it would be nicer to allow for empty pictures.) □

Example 2.5 We consider the alphabet $\Sigma = \{a, b\}$ and the set L of pictures of even height that have only a's in row $0, 2, 4, \ldots$ and only b's in rows $1, 3, \ldots$. Thus

$$L = ((\{a\} \ominus \{b\})^{\ominus+})^{\varobar+} = (\{a\}^{\varobar+} \ominus \{b\}^{\varobar+})^{\ominus+}.$$

Similarly to the two strategies (column-wise and line-wise, respectively) that the above "regular expressions" use to assemble L, we give two distinct PPPA that recognize this language.
Let $\mathfrak{A}_1 = (\{0,1,2\}, \Sigma, 0, \Delta_1, \{0\}, \sigma_1, \leq_1)$, with $\sigma_1(0) = \mathbf{r}$, $\sigma_1(1) = \sigma_1(2) = \mathbf{d}$, $1 \equiv_1 2 <_1 0$, and Δ_1 given by the following table:

0	a	1	0
1	b	2	0
2	a	1	0
1	b	$\top$	0

Then $\mathfrak{A}$ is a PPPA that scans pictures column-wise and recognizes L.
Let $\mathfrak{A}_2 = (\{0,1,2,3\}, \Sigma, 0, \Delta_2, \{0\}, \sigma_2, \leq_2)$, where $\sigma_2(q) = \mathbf{d}$ iff q is even, and the quasi-ordering is given by $3 \equiv_2 1 < 2 \equiv_2 0$, and Δ_2 is given by the following table:

0	a	2	1
1	a	2	1
1	a	2	$\top$
2	b	0	3
3	b	0	3
3	b	0	$\top$

Then $\mathfrak{A}_2$ is a PPPA that scans pictures line-wise and recognizes L, too. Apparently, L can be scanned both line-wise and column-wise. □

3 Regular Picture Languages and Automata

The main contributions of this paper are the following two theorems, stating that the class of regular picture languages is characterized by PPPA.

Theorem 3.1 Every regular picture language is recognized by some PPPA. □

Theorem 3.2 Every picture language recognized by some PPPA is regular. □

To prepare the proof of Theorem 3.1 we state a few lemmas.

Lemma 3.3 Let $\mathfrak{A}$ be a PPPA with $\sigma(q_0) = \mathbf{r}$. The set L of pictures of height 1 accepted by $\mathfrak{A}$ is a regular word language. □

Proof L is recognized by the non-deterministic finite state automaton (NFA) $(Q, \Sigma, q_0, \Delta', F)$, with transition relation

$$\Delta' = \{(q, a, r) \mid (q, a, \top, r) \in \Delta\},$$

which is easy to see. □

For a PPPA, the *chain height* is the maximal length n of a chain of the form $q_0 > q_1 > ... > q_{n-1}$. In other words, it is the height of the tree that $\leq$ induces on the set of $\equiv$-equivalence classes of Q.

Example 3.4 Every regular word language is recognized by a PPPA with chain height 1. □

Proof Let L be a regular word language. Choose an NFA $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F)$ that recognizes L. Then L is recognized by the PPPA $\mathfrak{A}' = (Q, \Sigma, q_0, \Delta', F, \sigma, \leq)$, where $\forall q \in Q : \sigma(q) = \mathbf{r}$, $\leq \, = Q \times Q$, and $\Delta' = \{(q, a, \top, r) \mid (q, a, r) \in \Delta\}$. □

From the above example it follows that for every letter $a \in \Sigma$, the singleton picture language $\{a\}$ is recognized by the two-state PPPA

$$(\{0, 1\}, \Sigma, 0, \{(0, a, \top, 1)\}, \{1\}, \sigma, \leq),$$

where $\sigma(0) = \sigma(1) = \mathbf{r}$ and $\leq \, = \{0, 1\} \times \{0, 1\}$.
The following lemma is this paper's technically most complicated one. Intuitively, it shows that the "outermost iteration direction" of a PPPA can be changed from **d** to **r** for the cost of a chain height increase by one.

Lemma 3.5 Let $\mathfrak{A}$ be a PPPA with $\sigma(q_0) = \mathbf{d}$. There is an equivalent PPPA $\mathfrak{A}'$ whose chain height is one heigher than that of $\mathfrak{A}$ and with $\sigma'(q_0') = \mathbf{r}$. □

Proof We may assume w.l.o.g. that Δ does not contain any tuple of the form $(q, a, \top, r)$ because such a tuple is irrelevant if $\sigma(q_0) = \mathbf{d}$.
Let q_0', f be two symbols not in Q, and choose $\mathfrak{A}' = (Q \cup \{q_0', f\}, \Sigma, q_0', \Delta', \{f\}, \sigma', \leq')$ with

$$\begin{array}{rcl} \sigma'(q) & := & \left\{ \begin{array}{cl} \sigma(q) & \text{if } q \in Q, \\ \mathbf{r} & \text{if } q \in \{q_0', f\}, \end{array} \right. \\ q \leq' q' & :\Leftrightarrow & q \leq q' \vee q' \in \{q_0', f\}, \\ \Delta' & := & \{(q', a, d', r') \mid \exists (q, a, d, r) \in \Delta : \\ & & \qquad (q' = q \vee (q' = q_0' \wedge q = q_0)) \\ & & \quad \wedge \ (d' = d \vee (d' = \top \wedge d \in F)) \\ & & \quad \wedge \ (r' = r \vee (r = \top \wedge r' = f))\}. \end{array}$$

One easily verifies that $\mathfrak{A}'$ is indeed a PPPA and its chain height is one heigher than that of $\mathfrak{A}$.
To see that $L(\mathfrak{A}) = L(\mathfrak{A}')$, let $P \in L(\mathfrak{A})$.
First we treat the case $|P| = 1$. For abbreviation, set $m = \underline{\overline{P}}$. Then there exist states $q_1, \ldots, q_m$ and configurations $\kappa_0, \ldots, \kappa_m$ such that $\kappa_0 \vdash \ldots \vdash \kappa_m$, and for every $i \in \{0, \ldots, m\}$ we have $\kappa_i = (P, (q_i, i, 0), \varepsilon, (\top, -1, 1))$, and $q_m \in F$. This implies $(q_i, P(i, 0), q_{i+1}, \top) \in \Delta$ for every $i \in \{0, \ldots, m-1\}$. Choose

$$\begin{array}{rcl} \kappa_0' & = & (P, (q_0', 0, 0), (\top, m, -1), \varepsilon), \\ \kappa_i' & = & (P, (q_i, i, 0), (\top, m, -1), (f, 0, 1)) \text{ for every } i \in \{1, \ldots m-1\}, \\ \kappa_m' & = & (P, (f, 0, 1), (\top, m, -1), \varepsilon). \end{array}$$

Then $\kappa_0' \vdash' \kappa_1'$ because $(q_0', P(0, 0), q_1, f) \in \Delta'$. For every $i \in \{1, \ldots m-2\}$ we have $\kappa_i' \vdash' \kappa_{i+1}'$ because $(q_i, P(i, 0), q_{i+1}, f) \in \Delta'$. Besides, $\kappa_{m-1}' \vdash' \kappa_m'$ because $(q_{m-1}, P(m-1, 0), \top, f) \in \Delta'$. Thus we have $\kappa_0' \vdash'^* \kappa_m'$ and thus $P \in L(\mathfrak{A}')$.
Now we treat the case $|P| > 1$. For abbreviation, let $m = \underline{\overline{P}} \cdot |P|$. By choice of P there exist P-configurations $\kappa_0, \ldots, \kappa_m$ such that $\kappa_0 \vdash \ldots \vdash \kappa_m$ and κ_0 is initial, κ_m is final,

$$\kappa_i = (P, (q_i, y_i, x_i), \alpha_i, \beta_i(\top, -1, |P|)).$$

The last step must pop the stack symbol $(q_m, \underline{\overline{P}}, 0)$ from the left stack (because $\sigma(q_m) = \sigma(q_0) = \mathbf{d}$), thus κ_{m-1} is of the form $(P, (q_{m-1}, \underline{\overline{P}} - 1, |P| - 1), (q_m, \underline{\overline{P}}, 0), (\top, -1, |P|)$. Among the $m - 1$ preceding steps, there are two critical ones. Firstly, the step k where the head leaves position $(\underline{\overline{P}} - 1, 0)$ and the symbol $(q_{m-1}, \underline{\overline{P}}, 0)$ is pushed onto the left stack, and secondly, the step l where the head leaves position $(0, |P| - 1)$. In the configurations $\kappa_k, \ldots, \kappa_{m-1}$, the bottom symbol of the left stack is $(q_m, \underline{\overline{P}}, 0)$. In the preceding configurations $\kappa_0, \ldots, \kappa_{k-1}$, the left stack is empty. Choose configurations $\kappa_0', \ldots, \kappa_m'$ as follows:

$$\begin{array}{rcl} \kappa_0' & = & (P, (q_0', 0, 0), (\top, \underline{\overline{P}}, -1), \varepsilon)) \\ \kappa_m' & = & (P, (f, \underline{\overline{P}}, 0), (\top, \underline{\overline{P}}, -1), \varepsilon)) \end{array}$$

and, for $i \in \{1, \ldots, m-1\}$,

$$\kappa_i' = \left\{ \begin{array}{lllll} (P, & (q_i, y_i, x_i), & \alpha_i'(\top, \underline{\overline{P}}, -1), & \beta_i) & \text{if } i < l \\ (P, & (q_i, y_i, x_i), & \alpha_i'(\top, \underline{\overline{P}}, -1), & \beta_i(f, 0, |P|)) & \text{if } i >= l, \end{array} \right.$$

where α_i' results from α_i by removing the symbol $(q_m, \overline{\underline{P}}, 0)$ if $i \geq k$. We wish to show $\kappa_0' \vdash' \ldots \vdash' \kappa_m'$.
In step k, the new automaton $\mathfrak{A}'$ does not push the symbol $(q_m, \overline{\underline{P}}, 0)$ onto the left (empty) stack as $\mathfrak{A}$ does in its run, but it rather exploits the transition $(q_m, P(y_m, x_m), \top, q_{m+1}) \in \Delta'$ and that the left top stack symbol has state $\top$.
In step l, the new automaton $\mathfrak{A}'$ does not see a right stack top symbol as $\mathfrak{A}$ does in its run, but it rather pushes the symbol $(f, 0, |P|)$ onto an empty stack.
In the remaining steps, $\mathfrak{A}'$ can work like $\mathfrak{A}$ does because the modification of the quasi-ordering $\leq'$ does not interfere with any of those steps. This shows that $P \in L(\mathfrak{A}')$ and thus completes the proof that $L(\mathfrak{A}) \subseteq L(\mathfrak{A}')$.
Conversely, let $P \in L(\mathfrak{A}')$. Again, set $m = \overline{\underline{P}} \cdot |P|$. There exist P-configurations $\kappa_0', \ldots, \kappa_m'$ such that κ_0' is initial, κ_m' is final, and $\kappa_0' \vdash' \ldots \vdash' \kappa_m'$.
We know that $\kappa_0' = (P, (q_0', 0, 0), (\top, \overline{\underline{P}}, -1), \varepsilon)$ and $\kappa_m' = (P, (f, 0, |P|), (\top, \overline{\underline{P}}, -1), \varepsilon)$. The remaining configurations $\kappa_1', \ldots, \kappa_{m-1}'$ cannot have states in $\{q_0', f\}$ because q_0' appears only in the first component of transitions and f only in the last. Thus κ_i' is, for $i \in \{1, \ldots, m-1\}$, of the form $\kappa_i' = (P, (q_i, y_i, x_i), \alpha_i(\top, \overline{\underline{P}}, -1), \beta_i)$ for $q_i \in Q$, $(y_i, x_i) \in \mathrm{dom}' P$, and (possibly empty) stack symbol words α_i, β_i.
The proof for the case $\overline{\underline{P}} = 1$ is simple and omitted.
Now we treat the case $\overline{\underline{P}} > 1$. Then the symbol $(f, 0, |P|)$ is popped from the right stack in the last step. Among the preceding $m - 1$ steps, there are two critical ones; firstly, step k where the head leaves position $(\overline{\underline{P}} - 1, 0)$ and secondly, step l where the head leaves position $(0, |P| - 1)$ and the symbol $(f, 0, |P|)$ is pushed onto the right stack. In step k, the automaton $\mathfrak{A}'$ applies the transition $(q_{k-1}, P(|P| - 1, 0), \top, q_k) \in \Delta'$. By definition of Δ' there must be a $\hat{q} \in F$ such that $(q_{k-1}, P(|P| - 1, 0), \hat{q}, q_k) \in \Delta$.
Choose

$$\begin{aligned}
\kappa_0 &= (P, (q_0, 0, 0), \varepsilon, (\top, \overline{\underline{P}}, -1)) \\
\kappa_m &= (P, (\hat{q}, \overline{\underline{P}}, 0), \varepsilon, (\top, \overline{\underline{P}}, -1)).
\end{aligned}$$

For every $i \in \{1, \ldots, m-1\}$ let

$$\kappa_i = \begin{cases} (P, \ (q_i, y_i, x_i), & \alpha_i, \ \beta_i') \quad \text{if } i < k \\ (P, \ (q_i, y_i, x_i), & \alpha_i(\hat{q}, \overline{\underline{P}}, 0), \ \beta_i') \quad \text{if } i \geq k \end{cases}$$

where β_i' results from β_i by removing the symbol $(f, 0, |P|)$ (if $i \geq l$).
We wish to show that $\kappa_0 \vdash \ldots \vdash \kappa_m$. In step k, where $\mathfrak{A}'$, in its run, sees a symbol with pseudo state $\top$ on its left stack, $\mathfrak{A}$ rather puts $(\hat{q}, \overline{\underline{P}}, 0)$ onto the empty left stack. In step l, where $\mathfrak{A}'$, in its run, pushes the symbol $(f, 0, |P|)$ onto the (empty) right stack, $\mathfrak{A}$ rather sees the symbol $(\top, -1, |P|)$ and continues without stack modification. In the remaining steps, $\mathfrak{A}$ works like $\mathfrak{A}'$. This shows that $P \in L(\mathfrak{A})$ and thus completes the proof. □

By symmetry, the above lemma remains true when the roles of **r** and **d** are interchanged. The same is true for the other lemmas that follow.

Lemma 3.6 Let $\mathfrak{A}_1, \mathfrak{A}_2$ be PPPAs with $\sigma_1(q_{01}) = \sigma_2(q_{02}) = \mathbf{r}$. Then $L(\mathfrak{A}_1) \cup L(\mathfrak{A}_2)$ is recognized by some PPPA whose chain height is the maximum of the chain heights of $\mathfrak{A}_1$, $\mathfrak{A}_2$. □

Proof Assume w.l.o.g. that Q_1 and Q_2 are disjoint. Let q_0 be a fresh state not in $Q_1 \cup Q_2$. Set $\mathfrak{A} = (Q_1 \cup Q_2 \cup \{q_0\}, \Sigma, q_0, \Delta, F_1 \cup F_2, \sigma, \leq)$, where

$$\begin{array}{rcl} \sigma(q) & := & \begin{cases} \sigma_1(q) & \text{if } q \in Q_1 \\ \sigma_2(q) & \text{if } q \in Q_2 \\ \mathbf{r} & \text{if } q = q_0, \end{cases} \\ q \leq q' & :\Leftrightarrow & q \leq_1 q' \vee q \leq_2 q' \vee q' \equiv_1 q_{01} \vee q' \equiv_2 q_{02} \vee q' = q_0, \\ \Delta & := & \Delta_1 \cup \Delta_2 \cup \{(q_0, a, d, r) \mid (q_{01}, a, d, r) \in \Delta_1 \vee (q_{02}, a, d, r) \in \Delta_2\}. \end{array}$$

One easily verifies that $\mathfrak{A}$ is a PPPA with the claimed chain height and that $L(\mathfrak{A}) = L(\mathfrak{A}_1) \cup L(\mathfrak{A}_2)$. □

Lemma 3.7 Let $\mathfrak{A}_1, \mathfrak{A}_2$ be PPPAs with $\sigma_1(q_{01}) = \sigma_2(q_{02}) = \mathbf{r}$. Then $L(\mathfrak{A}_1) \oplus L(\mathfrak{A}_2)$ is recognized by some PPPA whose chain height is the maximum of the chain heights of $\mathfrak{A}_1$, $\mathfrak{A}_2$. □

Proof Assume w.l.o.g. that Q_1 and Q_2 are disjoint. Furthermore we may assume[2] that $q_{01} \notin F_1$ and $q_{02} \notin F_2$. Set $\mathfrak{A} = (Q_1 \cup Q_2, \Sigma, q_{01}, \Delta, F_2, \sigma, \leq)$, where

$$\begin{array}{rcl} \sigma(q) & := & \begin{cases} \sigma_1(q) & \text{if } q \in Q_1 \\ \sigma_2(q) & \text{if } q \in Q_2 \end{cases} \\ q \leq q' & :\Leftrightarrow & q \leq_1 q' \vee q \leq_2 q' \vee q' \equiv_1 q_{01} \vee q' \equiv_2 q_{02}, \\ \Delta & := & \Delta_1 \cup \Delta_2 \cup \{(q, a, d, r) \mid q \in F_1 \wedge (q_{02}, a, d, r) \in \Delta_2\}. \end{array}$$

One easily verifies that $\mathfrak{A}$ is a PPPA with the claimed chain height. To see that $L(\mathfrak{A}) = L(\mathfrak{A}_1) \oplus L(\mathfrak{A}_2)$, let $P \in L(\mathfrak{A}_1) \oplus L(\mathfrak{A}_2)$, say $P = P_1 \oplus P_2$ with $P_1 \in L(\mathfrak{A}_1)$, $P_2 \in L(\mathfrak{A}_2)$. For abbreviation, let $m_1 = \underline{\overline{P}} \cdot |P_1|$. There exists P_1-configurations $\kappa_{1,0}, \dots, \kappa_{1,m_1}$ such that $\kappa_{1,0}$ is initial, κ_{1,m_1} is final, and $\kappa_{1,0} \vdash_1 \dots \vdash_1 \kappa_{1,m_1}$. For every $i \in \{0, \dots, m_1\}$ write

$$\kappa_{1,i} = (P_1, (q_{1,i}, y_{1,i}, x_{1,i}), \alpha_{1,i}, \beta_{1,i}),$$

for appropriate $x_{1,i}$, $y_{1,i}$, $\alpha_{1,i}$, $\beta_{1,i}$. Choose $\kappa_{2,0}, \dots, \kappa_{2,m_2}$ analogously with analogous naming of states, positions, and stack contents.
For abbreviation, set $m = m_1 + m_2$. For $i \in \{0, \dots, m\}$ define κ_i as follows:

$$\kappa_i = \begin{cases} (P, (q_0, 0, 0), (\top, \underline{\overline{P}}, -1), \varepsilon) & \text{if } i = 0, \\ (P, (q_{1,i}, y_{1,i}, x_{1,i}), \alpha_{1,i}, \beta_{1,i}) & \text{if } i \in \{1, \dots, m_1\}, \\ (P, (q_{2,i-m_1}, y_{2,i}, |P_1| + x_{2,i}), \alpha_{2,i}, \beta_{2,i}) & \text{if } i \in \{m_1 + 1, \dots, m_1 + m_2\}. \end{cases}$$

In order to show that $P \in L(\mathfrak{A})$, we have to show that $\kappa_0 \vdash \dots \vdash \kappa_m$. It is easy to see that $\kappa_0 \vdash \dots \vdash \kappa_{m_1}$ and $\kappa_{m_1+1} \vdash \dots \vdash \kappa_m$. By Remark 2.2, κ_{m_1} must be of the form $(P_1 \oplus P_2, (q_{1,m_1}, 0, |P_1|), (\top, \underline{\overline{P}}, -1), \varepsilon)$, thus it results from the starting configuration $\kappa'_{2,0}$ by prepending the picture P_1, which, as remarked in Remark 2.4, does not affect the operation of the PPPA.

[2]This is another point where it would easier to allow the empty picture, because then this assumption would not be necessary.

For the converse direction, let $P \in L(\mathfrak{A})$. For abbreviation, set $m = \underline{\overline{P}} \cdot |P|$. There exist P-configurations $\kappa_0, \ldots, \kappa_m$ such that $\kappa_0 \vdash \ldots \vdash \kappa_m$ and κ_0 is initial, κ_m is final. Say $\kappa_i = (P, (q_i, y_i, x_i), \alpha_i, \beta_i)$ for every i, then $q_0 = q_{01} \in Q_1$ and $q_m \in F_2 \subseteq Q_2$. Let n be minimal such that $q_n \in Q_2$. By Remark 2.2, κ_n must be of the form $(P, (q_n, 0, x), (\top, \underline{\overline{P}}, -1), \varepsilon)$ for some column index x. Since $q_{01} \notin F_1$ and $q_{02} \notin F_2$ we have $0 < x < |P|$. Choose pictures P_1, P_2 such that $P = P_1 \varobar P_2$ and $|P_1| = x$. An accepting run for P_1 in $\mathfrak{A}_1$ can be extracted from $\kappa_0 \vdash \ldots \vdash \kappa_{n-1}$ and similarly for P_2 from $\kappa_n \vdash \ldots \vdash \kappa_m$ using Remark 2.4. This shows that $P_1 \in L_1$ and $P_2 \in L_2$ and thus that $P \in L(\mathfrak{A}_1) \varobar L(\mathfrak{A}_2)$. This completes the proof. $\square$

Lemma 3.8 Let $\mathfrak{A}$ be a PPPA with $\sigma_1(q_0) = \mathbf{r}$. Then $L(\mathfrak{A})^{\varobar+}$ is recognized by some PPPA whose chain height equals that of $\mathfrak{A}$. $\square$

Proof Set $\mathfrak{A}' = (Q, \Sigma, q_0, \Delta', F, \sigma, \leq)$, where

$$\Delta' = \Delta \cup \{(q, a, d, r) \mid q \in F \backslash \{q_0\} \wedge (q_0, a, d, r) \in \Delta\}.$$

One easily verifies that $\mathfrak{A}$ is a PPPA with the claimed chain height. To see that $L(\mathfrak{A}') = L(\mathfrak{A})^{\varobar+}$, let $P \in L(\mathfrak{A})^{\varobar+}$. There exists $n \geq 1$ and pictures $P_1, \ldots, P_n \in L(\mathfrak{A})$ such that $P = P_1 \varobar \ldots \varobar P_n$. For all $i \in \{1, \ldots, n\}$ there exists $q_i \in F$ such that

$$(P_i, (q_0, 0, 0), (\top, \underline{\overline{P}}, -1), \varepsilon) \vdash^* (P_i, (q_i, 0, |P_i|), (\top, \underline{\overline{P}}, -1), \varepsilon).$$

Then by definition of Δ' we have

$$(P_i, (q_{i-1}, 0, 0), (\top, \underline{\overline{P}}, -1), \varepsilon) \vdash'^* (P_i, (q_i, 0, |P_i|), (\top, \underline{\overline{P}}, -1), \varepsilon)$$

for all $i \in \{1, \ldots, n\}$. For every $i \in \{0, \ldots, n\}$, set $j_i = |P_1| + \ldots + |P_i|$ and $\kappa_i = (P, (q_i, 0, j_i), (\top, \underline{\overline{P}}, -1), \varepsilon)$. Then by Remark 2.4, $\kappa_0 \vdash'^* \ldots \vdash'^* \kappa_n$ and κ_0 is initial and κ_n is final, hence $P \in L(\mathfrak{A}')$.

Conversely, let $P \in L(\mathfrak{A}')$. There exists $n \geq 1$ and configurations $\kappa_0, \ldots, \kappa_n$, $\kappa'_1, \ldots, \kappa'_n$ such that κ_0 is initial, κ_n is final, and $\kappa_0 \vdash^* \kappa_1$, and for every $i \in \{1, \ldots, n\}$ we have $\kappa_i \vdash \kappa'_i$ by application of a transition not in Δ and $\kappa'_i \vdash^* \kappa_{i+1}$ by application of transitions in Δ. The states of $\kappa_0, \ldots, \kappa_n$ are in $F \subseteq [q_0]$, so by Remark 2.2, κ_i is, for every i, of the form $(P, (q_i, 0, j_i), (\top, \underline{\overline{P}}, -1), \varepsilon)$ for some $q_i \in F$ such that $0 = j_0 < \ldots < j_n = |P|$. Choose pictures $P_1, \ldots, P_n$ such that $P = P_1 \varobar \ldots \varobar P_n$ and $|P_i| = j_i - j_{i-1}$ for every $i \in \{1, \ldots, n\}$. Then for every $i \in \{2, \ldots, n\}$ we have

$$(P_i, (q_{i-1}, 0, 0), (\top, \underline{\overline{P}}, -1), \varepsilon) \vdash'^* (P_i, (q_i, 0, 0), (\top, \underline{\overline{P}}, -1), \varepsilon),$$

where the first step is the only one that applies a transition not in Δ. Thus for every $i \in \{2, \ldots, n\}$ we have

$$(P_i, (q_0, 0, 0), (\top, \underline{\overline{P_i}}, -1), \varepsilon) \vdash^* (P_i, (q_i, 0, 0), (\top, \underline{\overline{P_i}}, -1), \varepsilon).$$

This is true also for $i = 1$, thus $P_i \in L(\mathfrak{A})$ for every $i \in \{1, \ldots, n\}$. This means $P = P_1 \varobar \ldots \varobar P_n \in L(\mathfrak{A})^{\varobar+}$, which completes the proof. $\square$

Proof Theorem 3.1 Let *PPPA* denote the class of PPPA-recognizable picture languages. By Example 3.4, *PPPA* contains all those singleton picture languages whose element has size 1×1. By Lemmas 3.5, 3.6, 3.7, and 3.8, *PPPA* is closed under union, column concatenation, and column closure. By symmetry, *PPPA* is also closed under row concatenation and row closure. This implies that REG $\subseteq$ *PPPA*. □

The proof shows in particular that the chain height of the PPPA corresponds to the nesting depth of row concatenation and -closure on one side and column concatenation and -closure on the other side: If that nesting depth for a regular expression is n, there is a PPPA with chain height n that recognizes the specified language. See Definition 4.6 and Remark 4.8. To prepare the proof of Theorem 3.2, we need another definition and a few lemmas.

Definition 3.9 Let $\mathfrak{A}$ be a PPPA with $\sigma(q_0) = \mathbf{r}$. Let $q_1, q_2 \in [q_0]$, $Q' \subseteq [q_0]$. Let P be a picture. We say that $\mathfrak{A}$ *consumes* P *from* q_1 *to* q_2 *via* Q' iff

$$(P, (q_1, 0, 0), (\top, \underline{\overline{P}}, -1), \varepsilon) \vdash^* (P, (q_2, 0, |P|), (\top, \underline{\overline{P}}, -1), \varepsilon)$$

in such a way that the intermediate states are not in $[q_0] \backslash Q'$. □

Note that for a PPPA $\mathfrak{A}$ with $\sigma(q_0) = \mathbf{r}$, we have

$$L(\mathfrak{A}) = \bigcup_{f \in F} \{P \mid \mathfrak{A} \text{ consumes } P \text{ from } q_0 \text{ to } f \text{ via } [q_0]\}.$$

Lemma 3.10 Let $\mathfrak{A}$ be a PPPA with $\sigma(q_0) = \mathbf{r}$, let $P \in L(\mathfrak{A})$. There exists $n \geq 1$ and $q_1, \ldots, q_n \in [q_0]$ and pictures $P_1, \ldots, P_n$ such that $P = P_1 \varobar \ldots \varobar P_n$ and for every $i \geq 1$, the PPPA $\mathfrak{A}$ consumes P_i from q_{i-1} to q_i via $\emptyset$. □

Proof Since $P \in L(\mathfrak{A})$, there is a run R that takes the initial configuration $(P, (q_0, 0, 0), (\top, \underline{\overline{P}}, -1), \varepsilon)$ to the final configuration $(P, (f, 0, |P|), (\top, \underline{\overline{P}}, -1), \varepsilon)$ for some state $f \in F$. Choose $n \geq 1$ and configurations $\kappa_0, \ldots, \kappa_n$ such that R reads $\kappa_0 \vdash^* \ldots \vdash^* \kappa_n$ and $\kappa_0, \ldots, \kappa_n$ are exactly those configurations of R whose state q_i is in $[q_0]$. (Note that $q_n = f$.)
By Remark 2.2, follows that, for every $i \in \{0, \ldots, n\}$, the configuration κ_i is of the form $(P, (q_i, 0, j_i), (\top, \underline{\overline{P}}, -1), \varepsilon)$. We have $0 = j_0 < \ldots < j_n = |P|$. Write $P = P_1 \varobar \ldots \varobar P_n$ such that $|P_i| = j_i - j_{i-1}$ for every $i \in \{1, \ldots n\}$. Then $\mathfrak{A}$ indeed consumes P_i from q_{i-1} to q_i via $\emptyset$. □

Lemma 3.11 Let $\mathfrak{A}$ be a PPPA with chain height > 1 and $\sigma(q_0) = \mathbf{r}$. Let $q_1, q_2 \in [q_0]$. Let L be the set of pictures that $\mathfrak{A}$ consumes from q_1 to q_2 via $\emptyset$. There exist two PPPAs $\mathfrak{A}_\mathbf{d}, \mathfrak{A}_\mathbf{r}$ with smaller chain height than $\mathfrak{A}$ and $L(\mathfrak{A}_\mathbf{d}) \cup L(\mathfrak{A}_\mathbf{r}) = L$. □

Proof Let f be a fresh symbol not in Q. Let $A_\mathbf{r}$ be the set of immediate $\leq$-successors q of q_0 with $\sigma(q) = \mathbf{r}$. Define $A_\mathbf{d}$ analogously.

Define $\mathfrak{A}_\mathbf{d} = (Q', \Sigma, q_1, \Delta_\mathbf{d}, \{f\}, \sigma_\mathbf{d}, \leq_\mathbf{d})$ and $\mathfrak{A}_\mathbf{r} = (Q', \Sigma, q_1, \Delta_\mathbf{r}, \{f\}, \sigma_\mathbf{r}, \leq_\mathbf{r})$ with $Q' = \{q_1, f\} \cup Q\backslash[q_0]$

$$
\begin{array}{rcl}
\sigma_\mathbf{d}(q) & := & \begin{cases} \sigma(q) & \text{if } q \in Q\backslash[q_0] \\ \mathbf{d} & \text{if } q \in \{q_1, f\}, \end{cases} \\
\sigma_\mathbf{r}(q) & := & \begin{cases} \sigma(q) & \text{if } q \in Q\backslash[q_0] \\ \mathbf{r} & \text{if } q \in \{q_1, f\}, \end{cases} \\
q \leq_\mathbf{d} q' & :\Leftrightarrow & q \leq q' < q_0 \vee ((\exists p \in A_\mathbf{d} : q \leq p) \wedge q' \in \{q_1, f\}), \\
q \leq_\mathbf{r} q' & :\Leftrightarrow & q \leq q' < q_0 \vee ((\exists p \in A_\mathbf{r} : q \leq p) \wedge q' \in \{q_1, f\}), \\
\Delta_\mathbf{d} & := & \quad \{(q, a, d, r) \in \Delta \mid q, d, r \notin [q_0]\} \\
& & \cup \ \{(q_1, a, d, r) \in \Delta \mid d, r \notin [q_0] \wedge (d < r \vee d = \top)\} \\
& & \cup \ \{(q, a, \top, f) \mid (q, a, \top, q_2) \in \Delta \wedge (q < q_0 \vee q = q_1)\} \\
\Delta_\mathbf{r} & = & \quad \{(q, a, d, r) \in \Delta \mid q, d, r \notin [q_0]\} \\
& & \cup \ \{(q_1, a, d, r) \in \Delta \mid d, r \notin [q_0] \wedge (r < d \vee r = \top)\} \\
& & \cup \ \{(q, a, \top, f) \mid (q, a, \top, q_2) \in \Delta \wedge (q < q_0 \vee q = q_1)\}
\end{array}
$$

$\mathfrak{A}_\mathbf{d}$ (or $\mathfrak{A}_\mathbf{r}$) simulates those runs of $\mathfrak{A}$ that start by moving the head rightwards (or downwards, respectively).
If $\mathfrak{A}_\mathbf{d}$ is started on some input picture P, it simulates any such run of $\mathfrak{A}$ started in q_1 until $\mathfrak{A}$ tries to push a stack symbol with state in $[q_0]$ onto the right stack. If this state is $\neq q_2$, the PPPA $\mathfrak{A}_\mathbf{d}$ cannot proceed and thus does not accept. Otherwise it pushes a stack symbol with state f onto the right stack and proceeds like $\mathfrak{A}$ until $\mathfrak{A}$ pops that stack symbol. Then $\mathfrak{A}_\mathbf{d}$ cannot proceed, so P is accepted iff it has been completely scanned. □

Now we are ready to show that the language accepted by a PPPA is regular.

Proof Theorem 3.2 We argue by induction on the chain height of PPPA. Let $\mathfrak{A}$ be a PPPA of height 1. W.l.o.g. assume $\sigma(q_0) = \mathbf{r}$. (Otherwise consider $L(\mathfrak{A})^\top$.) Then $\mathfrak{A}$ cannot make any steps on pictures of height ≥ 2, so $L(\mathfrak{A})$ is regular by Lemma 3.3.
For the induction step, let $\mathfrak{A}$ be a PPPA of chain height > 1. We may assume as induction hypothesis that every picture language recognized by a PPPA with smaller chain height is regular. Again we assume w.l.o.g. that $\sigma(q_0) = \mathbf{r}$. Choose pairwise distinct $q_1, \ldots, q_n$ such that $[q_0] = \{q_0, \ldots, q_n\}$. For $i, j \leq n$ and $k \leq n+1$ we define the picture language L_{ijk} as the set of pictures P such that $\mathfrak{A}$ consumes P from q_i to q_j via $\{q_0, \ldots, q_{k-1}\}$. By Lemma 3.11 and induction hypothesis, L_{ij0} is regular for every i, j. We claim that for every $i, j, k \leq n$ we have

$$L_{i,j,k+1} = L_{ijk} \cup (L_{ikk} \varobar L_{kkj}) \cup (L_{ikk} \varobar L_{kkk}^{\varobar+} \varobar L_{kkj}) \tag{1}$$

The direction "⊆" of Equation (1) follows from Lemma 3.10. The other direction is simple. We may conclude by induction over k that $L_{i,j,k}$ is regular for every i, j and every k. It follows that $L(\mathfrak{A}) = \bigcup\{L_{0,j,n+1} \mid q_j \in F\}$ is regular, which completes the proof. □

Since the class of regular picture languages is closed under rotation, we can conclude the following from Theorems 3.1 and 3.2, which is not obvious from the definition of PPPA.

Corollary 3.12 The class of PPPA-recognizable picture languages is closed under rotation. □

Since the class of regular picture languages is not closed under intersection, we can conclude similarly:

Corollary 3.13 The class of PPPA-recognizable picture languages is not closed under intersection (unless the alphabet Σ is a singleton). □

A simple example for a non-regular intersection of two regular picture languages is the set of all those pictures P over alphabet $\{a, b\}$ for which there is one row i and one column j such that any cell carries a b iff it is in that row i or in that column j.
In the same way we may conclude:

Corollary 3.14 Every PPPA-recognizable picture language is tiling-recognizable. □

4 Regular Picture Languages and Periodicity

We consider the set $\mathbb{N} = \{1, 2, \ldots\}$ of positive integers. A set $N \subseteq \mathbb{N}$ is *ultimately periodic* iff there exist $p, t \geq 1$ such that $\forall n \geq t : n \in N \leftrightarrow n + p \in N$.

Remark 4.1 Let $\varphi : \Sigma^+ \to \mathbb{N}$, $w \mapsto |w|$. If L is a regular word language, then $\varphi(L) \subseteq \mathbb{N}$ is ultimately periodic. Conversely, if $N \subseteq \mathbb{N}$ is ultimately periodic, then $\varphi^{-1}(N) \subseteq \Sigma^+$ is a regular word language □

We denote the powerset of a set M by 2^M. We need the following lemma that has nothing to do with pictures and has probably been proved before by other authors.

Lemma 4.2 Let $\psi : \Gamma \to 2^{\{0\}^+}$ be a mapping such that $\psi(a)$ is a regular word language over $\{0\}$ for every $a \in \Sigma$. We lift ψ to a map $\Gamma^+ \to 2^{\{0\}^+}$ as usual, i.e., $\psi(a_1 \ldots a_n) := \psi(a_1) \ldots \psi(a_n)$. Let $L \subseteq \{0\}^+$ be regular. Then $\psi^{-1}(L)$ is regular. □

Proof The claim follows from the closure of the class of regular word languages under rational transductions and inverse rational transductions. □

The following fact is similar to Remark 4.1 and has been shown in [Mat97]. We define a mapping size : $\Sigma^{+,+} \to \mathbb{N} \times \mathbb{N}$, $P \mapsto (\underline{P}, |P|)$ and lift this mapping to picture languages as usual.

Remark 4.3 If L is a regular picture language, then $\text{size}(L) \subseteq \mathbb{N} \times \mathbb{N}$ is a finite union of Cartesian products of ultimately periodic subsets of $\mathbb{N}$. Conversely, if $\mathcal{N} \subseteq \mathbb{N} \times \mathbb{N}$ is a finite union of Cartesian products of ultimately periodic subsets of $\mathbb{N}$, then $\text{size}^{-1}(\mathcal{N})$ is a regular picture language. □

The following is a simple consequence.

Remark 4.4 Consider the mapping length : $\Sigma^{+,+} \to \mathbb{N}$, $P \mapsto |P|$ and lift this mapping to picture languages as usual. Then $\text{length}(L)$ is ultimately periodic for every regular picture language L. □

Let Γ be another finite alphabet. A *regular substitution from* Γ *into* Σ is a mapping φ from Γ into the class of regular picture languages over Σ. Such a mapping is lifted to the set Γ^+ of non-empty words as follows:

$$\forall a_1, \ldots, a_n \in \Gamma : \varphi(a_1 \ldots a_n) := \varphi(a_1) \oplus \ldots \oplus \varphi(a_n).$$

This mapping is further lifted to word languages as usual. Obviously, if M is a regular word language over Γ, then $\varphi(M)$ is a regular picture language.

Remark 4.5 Let $\varphi : \Gamma \to \Sigma^{+,+}$ be a regular substitution. Let $N \subseteq \mathbb{N}$ be ultimately periodic. Define another regular substitution $\varphi' : \Gamma \to \Sigma^{+,+}$, $a \mapsto \varphi(a) \cap \text{height}^{-1}(N)$. Then $\varphi(M) \cap \text{height}^{-1}(N) = \varphi'(M)$ for every word language M over Γ. □

For a picture P, its transposed is denoted $P^\top$. As usual, $L^\top = \{P^\top \mid P \in L\}$ for every picture language L.

Definition 4.6 For $n \geq 1$ and a finite alphabet Σ, the class $\text{REG}_{\mathbf{r},n}(\Sigma)$ of picture languages over Σ is inductively defined as follows:

- $\text{REG}_{\mathbf{r},1}(\Sigma)$ is the set of regular word languages over Σ.
- $\text{REG}_{\mathbf{r},n+1}(\Sigma)$ is the set of picture languages for which there is a regular word language over some alphabet Γ and a regular substitution $\varphi : \Gamma \to \Sigma^{+,+}$ such that $\varphi(a)^\top$ is in $\text{REG}_{\mathbf{r},n}(\Gamma)$ for all $a \in \Gamma$.

For every $n \geq 1$, let $\text{REG}_{\mathbf{d},n}(\Sigma)$ denote the set of picture languages whose transposed is in $\text{REG}_{\mathbf{r},n}(\Sigma)$. The dependency to Σ is usually dropped when it is clear from the context. □

Equivalently, one may define that $\text{REG}_{\mathbf{r},n+1}$ is the smallest superclass of $\text{REG}_{\mathbf{d},n}$ that is closed under union, row concatenation, and row closure.
We will call $\text{REG}_{\mathbf{r},n}$ the *n-th level of the concatenation alternation hierarchy.* Obviously, $\text{REG}_{\mathbf{r},n} \subseteq \text{REG}_{\mathbf{r},n+1}$ for every $n \geq 1$, and $\bigcup_{n\geq 1} \text{REG}_{\mathbf{r},n} = \text{REG}$.

Example 4.7 The language L of Example 2.5 is in $\text{REG}_{\mathbf{d},2} \cap \text{REG}_{\mathbf{r},2}$.
To see that it is in $\text{REG}_{\mathbf{d},2}$, consider $\varphi_1 : \{0,1\} \to 2^{\{a,b\}^{+,+}}$ with $\varphi_1(0) = (\{a\}^+)^\top$ and $\varphi_1(1) = (\{b\}^+)^\top$. Then $L = \varphi_1(\{01\}^+)^\top$.
To see that it is in $\text{REG}_{\mathbf{r},2}$, consider $\varphi_2 : \{0\} \to 2^{\{a,b\}^{+,+}}$ with $\varphi_2(0) = (\{ab\}^+)^\top$. Then $L = \varphi_2(\{0\}^+)$. □

Remark 4.8 The proofs in Section 3 show that a language is recognized by a PPPA with chain-height n and $\sigma(q_0) = \mathbf{r}$ iff it is in $\text{REG}_{\mathbf{r},n}$. By symmetry, the same is true for $\mathbf{d}$ instead of $\mathbf{r}$. □

The above remark is important because it indicates that both the notion of chain-height and of the concatenation hierarchy are somewhat natural. It is pretty clear that the concatenation hierarchy is strict, i.e., we conjecture that $\text{REG}_{\mathbf{r},n} \subsetneq \text{REG}_{\mathbf{r},n+1}$ for every n. A proof for this is in preparation.

In the following, we will need the levels of the concatenation hierarchy for an inductive argument. We aim to show Proposition 4.11 stating, roughly speaking, that the partialness of the concatenation can, on the language level, be overcome. We start with a few more definitions.

Let L be a picture language over Σ. We say that $\mathrm{REG}_{\mathbf{r},n}$ is *intersection resistant* if for all picture languages $L \in \mathrm{REG}_{\mathbf{r},n}$ and all ultimately periodic sets $N_1, N_2 \subseteq \mathbb{N}$, the picture language $L \cap \mathrm{size}^{-1}(N_1 \times N_2)$ is in $\mathrm{REG}_{\mathbf{r},n}$, too.

Lemma 4.9 $\mathrm{REG}_{\mathbf{r},n}$ is intersection resistant for every $n \geq 1$. $\square$

Proof We argue by induction on n. The fact that $\mathrm{REG}_{\mathbf{r},1}$ is intersection resistant follows from Remark 4.1 and the closure of the class of regular word languages under intersection. For the induction step, let $n \geq 1$ and assume that $\mathrm{REG}_{\mathbf{r},n}$ is intersection resistant.

Let $L \in \mathrm{REG}_{\mathbf{r},n+1}(\Sigma)$. There exists a finite alphabet Γ and a regular substitution $\varphi : \Gamma \to \mathrm{REG}_{\mathbf{d},n}(\Sigma)$ and a regular word language $M \subseteq \Gamma^+$ such that $L = \varphi(M)$.

Let $N_1, N_2 \subseteq \mathbb{N}$ be ultimately periodic.

For every $X \subseteq \Gamma$ we define a regular substitution φ_X from X into Γ as follows

$$\varphi_X(a) = \varphi(a) \cap \mathrm{height}^{-1}\left(N_1 \cap \bigcap_{a \in X} \mathrm{height}(\varphi(a))\right).$$

By induction hypothesis, φ_X is indeed a regular substitution. For every $X \subseteq \Gamma$ we define $\psi_X : X \to 2^{\{0\}^+}, a \mapsto \{0^n \mid n \in \mathrm{length}(\varphi_X(a))\}$ and extend this mapping to $X^+ \to 2^{\{0\}^+}$ as usual. Besides, we define $\pi : \{0\}^+ \to \mathbb{N}$, $w \mapsto |w|$. Then $\psi_X^{-1}(\pi^{-1}(N_2)) = \{a_1 \ldots a_n \in X^+ \mid \exists P_1 {\in} \varphi(a_i) \ldots \exists P_n {\in} \varphi(a_n) : |P_1 \ldots P_n| \in N_2\} = \{w \in X^+ \mid \mathrm{length}(\varphi_X(w)) \in N_2\}$ is a regular word language by Remark 4.1 and because $\varphi_X(w)$ is a regular word language for every $X \subseteq \Gamma$ and every $w \in X^+$ by Remark 4.4.

Define $K_X := \varphi_X(M) \cap \mathrm{length}^{-1}(N_2)$. Then $K_X = \varphi_X(M \cap \psi_X^{-1}(\pi^{-1}(N_2)))$. Since $M \cap \psi_X^{-1}(\pi^{-1}(N_2))$ is a regular word language (because of the closure properties of that class), $K_X \in \mathrm{REG}_{\mathbf{r},n+1}(\Sigma)$.

Then

$$\begin{aligned} & L \cap \mathrm{size}^{-1}(N_1 \times N_2) \\ &= \textstyle\bigcup_{X \subseteq \Gamma} \big(\varphi(M \cap X^+) \cap \mathrm{height}^{-1}(N_1) \cap \mathrm{length}^{-1}(N_2)\big) \\ &= \textstyle\bigcup_{X \subseteq \Gamma} \big(\varphi_X(M) \cap \mathrm{length}^{-1}(N_2)\big) \\ &= \textstyle\bigcup_{X \subseteq \Gamma} K_X. \end{aligned}$$

(For the second equation, direction "$\subseteq$", see Remark 4.5; for direction "$\supseteq$" note $\varphi_X(M) \subseteq \varphi(M \cap X^+)$.) Thus $L \cap \mathrm{size}^{-1}(N_1 \times N_2)$ is a finite union of languages in $\mathrm{REG}_{\mathbf{r},n+1}$ and therefore itself in that class. This completes the proof. $\square$

Now we define a partial binary operation $\varobar'$ on picture languages called the *strict column concatenation.*

Let L_1, L_2 be picture languages. $L_1 \varobar' L_2$ is defined as $L_1 \varobar L_2$ iff

$$\begin{aligned} \forall P_1 \in L_1 \, \exists P_2 \in L_2 : &\quad \overline{P_1} = \overline{P_2}, \\ \forall P_2 \in L_2 \, \exists P_1 \in L_1 : &\quad \underline{P_1} = \underline{P_2}. \end{aligned}$$

Otherwise, $L_1 \oplus' L_2$ is undefined. The strict row concatenation $\ominus'$ is defined analogously, referring to width rather than height.

Definition 4.10 The class of strictly regular picture languages is the smallest class that contains all singleton languages whose element is of size 1×1 and that is closed under union, strict column concatenation, strict row concatenation, column closure, and row closure. □

Intuitively, a picture language is strictly regular iff it can be assembled without exploiting the partialness of the concatenation: whatever picture has been assembled before must contribute to the result. Obviously, every strictly regular picture is regular. The converse is also true:

Proposition 4.11 Every regular picture language is strictly regular. □

Proof We show by induction over n that every picture language in $\mathrm{REG}_{\mathbf{r},n}$ is strictly regular. For $n = 1$, this is immediate.
Now let $n \geq 1$ and assume that every picture language in $\mathrm{REG}_{\mathbf{r},n}$ is strictly regular. Let $L \in \mathrm{REG}_{\mathbf{r},n+1}$.
There exists a finite alphabet Γ and a regular substitution $\varphi : \Gamma \to \Sigma^{+,+}$ such that $\varphi(a) \in \mathrm{REG}_{\mathbf{d},n}(\Sigma)$ for every $a \in \Gamma$, and $\varphi(M) = L$.
By Remark 4.3, for every $a \in \Sigma$ there is an ultimately periodic $N_a \subseteq \mathbb{N}$ such that $\mathrm{height}(\varphi(a)) = N_a$. For every $X \subseteq \Gamma$, the set $N_X := \bigcap_{a\in X} N_a$ is ultimately periodic. For every $X \subseteq \Gamma$, define a regular substitution $\varphi_X : X \to \Sigma^{+,+}$, $a \mapsto \varphi(a) \cap \mathrm{height}^{-1}(N_X)$.
By Lemma 4.9, $\varphi_X(a)$ is in $\mathrm{REG}_{\mathbf{d},n}$ for every $a \in X \subseteq \Gamma$. By induction hypothesis, $K_X := \varphi_X(\Gamma^+ \cap M)$ is the transposed of a strictly regular picture language and hence itself strictly regular.
Now we have

$$\begin{aligned} L &= \varphi(M) \\ &= \textstyle\bigcup_{X\subseteq\Gamma}(\varphi(\Gamma^+ \cap M) \cap \bigcap_{a\in X} \mathrm{height}^{-1}(N_a)) \\ &= \textstyle\bigcup_{X\subseteq\Gamma}(\varphi(\Gamma^+ \cap M) \cap \mathrm{height}^{-1}(N_X)) \\ &= \textstyle\bigcup_{X\subseteq\Gamma} \varphi_X(\Gamma^+ \cap M) \\ &= \textstyle\bigcup_{X\subseteq\Gamma} K_X. \end{aligned}$$

(For the second equation, direction "$\supseteq$", consider for a word $w \in \Gamma^+$ the set X of all of w's letters. For the fourth equation, see Remark 4.5.) So L is a finite union of strictly regular picture languages and thus strictly regular. This completes the proof. □

Remark 4.12 Speaking in syntactic terms, we have considered a subclass of the class of regular expressions by forbidding expressions that exploit the partialness of the concatenations.
The above proof shows that a little more than stated in Proposition 4.11: it shows that for every regular expression, that subclass contains an equivalent one *with the same concatenation alternation depth.* □

For every picture P, define $\text{front}(P) = P(0,0)\dots P(0,|P|-1)$ as the word in its top row. As usual, we define $\text{front}(L) = \{\text{front}(P) \mid P \in L\}$ for a picture language L.
The following corollary is not completely trivial because in general the inclusion

$$\text{front}(L_1 \mathbin{⦶} L_2) \subseteq \text{front}(L_1)\text{front}(L_2)$$

is proper due to the partialness of the column concatenation.

Corollary 4.13 For every regular picture language L, the word language $\text{front}(L)$ is regular. □

Proof Let L_1, L_2 be picture languages. Then $\text{front}(L_1 \cup L_2) = \text{front}(L_1) \cup \text{front}(L_2)$ and $\text{front}(L_1^{⦶+}) = \text{front}(L_1)^+$, and $\text{front}(L_1^{\ominus+}) = \text{front}(L_1)$. Moreover, if $L = L_1 \mathbin{⦶'} L_2$, then $\text{front}(L) = \text{front}(L_1)\text{front}(L_2)$. Besides, if $L = L_1 \ominus' L_2$, then $\text{front}(L) = \text{front}(L_1)$. Using this observation, the claim follows by simple induction from Proposition 4.11. □

This last corollary complements a result of [LS97] stating that the *context-sensitive* word languages are exactly those of the form $\text{front}(L)$ for a tiling-recognizable picture language L. In other words, the operation front on tiling-recognizable picture languages leaves the world of regularity whereas it does not on regular picture languages.
This might be interpreted as another indication that the class of regular picture language is robust and naturally corresponds to regular world known from formal word languages.
The world of tree languages is another natural and well-studied playground to transfer investigations of word languages. For a tree, $\text{front}(t)$ denotes the sequence of leaf-symbols, so in this world, too, front denotes an operation into the set of words. It is well-known that the context-free word languages are exactly those of the form $\text{front}(L)$ for a regular tree language (where regularity is pretty undoubtedly the "right" definition.)

5 Conclusion

We have studied the class of regular picture languages and found a non-deterministic automaton model that captures this class. The following questions may be the subject of further investigations. Apart from that we have shown that the partialness of the two concatenations is irrelevant for the assembly of regular picture languages.
The two areas have in common that the concatenation alternation hierarchy naturally pops up, indicating that this hierarchy deserves investigation by itself.

- For practical purposes, a deterministic automaton model for the class REG of regular picture languages would be desirable. However, since REG is not closed under complement (not even under intersection), and because a deterministic automaton model typically implies closure under complement, there is little or no hope that this is possible.

  Is there some convenient sub- or superclass of REG that allows for such a model and preserves the nice closure- and decidability properties of REG?

- Can PPPA be used to construct fast algorithms for 2-dimensional pattern matching and model checking? Maybe one can borrow determinization and minimization techniques from ordinary NFA and adapt them to reduce the amount of non-determinism needed.
- There are picture languages such as the set of squares that are very simple but not regular. One possible way out has been presented in [Mat97] with the definition of the class of regular picture languages with operators.

  How must the definition of PPPA be adapted to capture or subsume that class in expressive power?
- Is the concatenation alternation hierarchy strict? I conjecture that the answer is yes and that the proof is fairly easy.
- Compare the PPPA with the pushdown and queue automata of [AM05]. These approaches have something in common, so maybe there is some uniform way to investigate them.
- Most of the papers about picture languages hardly exploit the limitation to two dimensions, and this one is no exception. With a minor increase of notational effort, it should be straightforward to transfer the results of this paper to multi-dimensional "tensor languages", and likewise those of [GRST96, LS94, Mat97] and other papers.

6 Acknowledgments

I thank Thomas Wilke for his advice and support, as well as for the opportunity to publish this paper as a technical report. I owe gratitude to my girlfriend Kristina Brandstädter for her patience while I am physically or mentally absent doing theoretical computer science.

References

[AGM04] Marcella Anselmo, Dora Giammarresi, and Maria Madonia. Regular expressions for two-dimensional languages over one-letter alphabet. In Cristian Calude, Elena Calude, and Michael J. Dinneen, editors, *Developments in Language Theory*, volume 3340 of *Lecture Notes in Computer Science*, pages 63–75. Springer, 2004.

[AGM05] Marcella Anselmo, Dora Giammarresi, and Maria Madonia. New operations and regular expressions for two-dimensional languages over one-letter alphabet. *Theor. Comput. Sci.*, 340(1):408–431, 2005.

[AM05] Marcella Anselmo and Maria Madonia. Simulating two-dimensional recognizability by pushdown and queue automata. In Jacques Farré, Igor Litovsky, and Sylvain Schmitz, editors, *CIAA*, volume 3845 of *Lecture Notes in Computer Science*, pages 43–53. Springer, 2005.

[GR96] D. Giammarresi and A. Restivo. Two-dimensional languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Language Theory*, volume III, pages 125–267. Springer-Verlag, New York, 1996.

[GRST96] D. Giammarresi, A. Restivo, S. Seibert, and W. Thomas. Monadic second-order logic and recognizability by tiling systems. *Information and Computation*, 125:32–45, 1996.

[IN77] K. Inoue and A. Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, 13:95–121, 1977.

[LS94] M. Latteux and D. Simplot. Recognizable picture languages and domino tiling. Internal Report IT-94-264, Laboratoire d'Informatique Fondamentale de Lille, Université de Lille, France, 1994.

[LS97] M. Latteux and D. Simplot. Context-sensitive string languages and recognizable picture languages. *Information and Computation*, 138:160–169, 1997.

[Mat95] O. Matz. *Klassifizierung von Bildsprachen mit rationalen Ausdrücken, Grammatiken und Logik-Formeln.* Diploma thesis, Christian-Albrechts-Universität Kiel, 1995. (in German).

[Mat97] O. Matz. Regular expressions and context-free grammars for picture languages. In Rüdiger Reischuk and Michel Morvan, editors, *Symposium on Theoretical Aspects of Computer Science 1997*, volume 1200 of *Lecture Notes in Computer Science*, pages 283–294, Lübeck, 1997. BRICS, Springer.

[Mat98] O. Matz. On piecewise testable, starfree, and recognizable picture languages. In Maurice Nivat, editor, *Foundations of Software Science and Computation Structures*, volume 1378 of *Lecture Notes in Computer Science*, pages 203–210. Springer, 1998.

[Rei98] K. Reinhardt. On some recognizable picture-languages. In Luboš Brim, Jozef Gruska, and Jiří Zlatuška, editors, *Mathematical Foundations of Computer Science*, volume 1450 of *Lecture Notes in Computer Science*, pages 760–770. Springer, 1998.

[Wil97] Th. Wilke. Star-free picture expressions are strictly weaker than first-order logic. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *Automata, Languages and Programming*, volume 1256, pages 347–357. Springer, Bologna, Italy, 1997.